POWER-AWARE HYBRID-DYNAMICAL APPROACH TO COVERAGE CONTROL IN MULTI-ROBOT SYSTEMS

A Thesis
Presented to
The Academic Faculty

By

Mark Olsen

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical Engineering

Georgia Institute of Technology

May 2018

POWER-AWARE HYBRID-DYNAMICAL APPROACH TO COVERAGE CONTROL IN MULTI-ROBOT SYSTEMS

Approved by:

Dr. Magnus Egerstedt, Advisor
School of Electrical Engineering
Georgia Institute of Technology

Dr. Yorai Wardi
School of Electrical Engineering
Georgia Institute of Technology

Dr. Samuel Coogan
School of Electrical Engineering
Georgia Institute of Technology

Date Approved: April 27, 2018

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Magnus Egerstedt, for opening my eyes to the fascinating area of swarm robotics and guiding my research over the past year. I also thank the members of the GRITS Lab for their continued help through this research effort, in particular Siddharth Mayya for his constant support as a mentor. I also gratefully acknowledge my employer, Sandia National Laboratories, for supporting my education at Georgia Tech. Finally, I thank my Mom, Dad, and Kellie for their loving support and encouragement over the last two years.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

This thesis develops an algorithm which allows robots in a multi-robot team to optimize for battery power while performing coverage control so as to maximize the mission life of the multi-robot team. We envision a scenario where robots with limited battery supply are executing the well known Lloyd's algorithm in order to effectively cover a certain region. We perform a trade-off between the distance of a robot from the centroid of its Voronoi cell, and the energy required to traverse that distance. In order to execute this trade-off two different strategies are presented – in one case, the reduction in cost due to coverage is compared against the energy required to traverse the distance to the centroid, and using a user-defined threshold, the decision is made. Then, a more sophisticated algorithm is used to perform the trade-off where the robots solves a switch-time optimization problem to decide whether it should move or it should stay.

We demonstrate that the developed algorithm's strategy allows the robots to operate for longer periods of time without recharging, while achieving the desired trade-off between coverage and battery consumption, by comparing the performance against robots executing a canonical coverage control algorithm. The optimization framework developed in this thesis can be used to qualitatively perform a trade-off between the level of coverage needed for a certain use-case and the constraints on the energy consumption of the robots. The relative importance of coverage and energy is left to the designer as a parameter that can be fit to the application under consideration. The developed methods are completely decentralized and only use information already known to the robots while executing the coverage control algorithm. Further, the algorithm does not require any additional communication between the agents.

CHAPTER 1
INTRODUCTION

Over the last decade, owing to advances in computational and sensory technologies, multi-robot systems are being deployed in the real-world to perform various tasks (for e.g., see [1, 2]). This is in part due to their ability to robustly and reliably operate in varied environments such as disaster zones [3], urban regions [4], and underwater environments [5].

A particularly well-studied multi-robot application is coverage control where a team of robots spread themselves in an area so as to satisfactorily cover regions of interest, e.g., [6]. Coverage control is applied in situations where mobile sensor nodes need to position themselves to detect relevant features and events in the region, or where robots need to effectively provide surveillance of a region by spreading across it [7]. Such operations are typically enabled by letting each robot be in charge of the part of the domain that is closest to it, and using this to define an overall quality of coverage. Given this formulation, the robots then execute a control algorithm in order to increase the overall quality of coverage [8].

This thesis is motivated by the fact that many coverage applications require the ability to persistently monitor a region for an extended period of time causing the rate of energy consumption to significantly impact coverage quality. To this end, we develop a strategy by which both the primary goals of coverage and energy can be balanced to improve upon already existing methods of coverage control such as in [8].

The outline of this thesis is as follows. This chapter gives a general introduction to the power-aware coverage problem (Section 1.1). This is followed by Chapter 2 where the technical background of the coverage problem is introduced (Section 2.1) as well as the robot's dynamics and energy consumption model (Section 2.2). In Chapter 3 this coverage and energy model are then used to develop algorithms that will be used to indirectly

reduce a global cost function (Section 3.1), where we introduce Threshold-Based Switching (Section 3.2) and Switch-Time Optimization (Section 3.3). In Chapter 4 the developed algorithms are verified in simulation by performing two experiments (Sections 4.1 and 4.2).

1.1 Problem Overview

Surveillance is an essential part of many multi-robot applications such as search and rescue missions [9], border patrol and security systems [10], and disaster relief zones [3]. These coverage applications involve the use of robots to sustain coverage with a finite supply of energy for an extended period of time. When a robot loses power, or must leave a region to replenish energy, a loss of surveillance occurs. This thesis looks to extend the period of time robots can provide coverage, and therefore increasing the quality of coverage, by creating an algorithm for movement that aims to balance the improvement in coverage with the cost associated in achieving it.

1.1.1 Problem Motivation

The coverage task requires the robots to move through the domain, typically based on a gradient descent method applied on the underlying cost function, such as in [6]. Owing to the fact that mobility in robotic systems is one of the biggest sources of power consumption [11], this can limit the ability of the multi-robot system to operate in external environments for long periods of time, as was observed in [12]. This is due to the fact that mobile robots are mainly battery-powered with a fixed amount of on-board energy and thus, the time for which the team can operate without the need to recharge is inversely proportional to the power consumed by the robots. Furthermore, recharging operations, as in [13], can incur significant penalties on the effectiveness of the tasks performed especially in remote environments. Thus, there is a need to reduce the frequency at which robots replenish their batteries while executing the coverage control algorithm [11].

1.1.2 Solution Strategy

Motivated by these issues, this thesis develops a motion strategy which allows each robot in a multi-robot system to balance its coverage objectives against the amount of expendable energy it has. At regular intervals of time, each robot evaluates the potential reduction in coverage cost that would result from the execution of the coverage control algorithm, and compares this against the amount of energy that will be consumed as a result of the motion. If the potential benefits of better coverage outweigh the reduction in available energy, the robot nominally executes the coverage algorithm. Conversely, if the effective reduction in the coverage cost does not seem commensurate with the energy required to move to a new location, the robot decides to *not* move. By doing so, the robots reduce their energy consumption, and ensure that they can operate for longer periods of time on a single charge.

In order to perform this trade-off, we use ideas from hybrid systems theory [14], and represent the robot as a hybrid dynamical system operating in two distinct modes – it either nominally moves according to the coverage control algorithm or remains stationary. For a given time horizon, we formulate a cost function which captures the reduction in coverage cost due to the motion of the robot, and the reduction in battery energy incurred due to this motion.

The first proposed strategy simply thresholds the two costs to decide whether the robot should operate nominally or remain stationary for a certain time horizon. The thresholding can be chosen in order to achieve a certain trade-off between the quality of coverage and the energy consumed by the robots in a certain amount of time. The second proposed strategy uses a switch-time optimization algorithm, such as in [15], to compute the optimal time to switch from one mode to another in a given time horizon. For simplicity, these algorithms are implemented in a decentralized manner to keep the robots from needing to share information with each other during execution.

CHAPTER 2
BACKGROUND THEORY

As described in Chapter 1, the goal of this thesis is to devise a strategy which allows robots in a multi-robot system to balance their coverage objectives against the energy expenditure incurred due to the motion. In order to do this systematically, we introduce cost functions which can allow for a comparison, and subsequently an optimization, between covering a region effectively, and conserving energy. To that end, this section introduces the necessary background theory required to develop such an optimization framework. Section 2.1 introduces Lloyd's algorithm, a canonical and widely-used algorithm which allows agents to cover a region effectively based on an underlying distribution of the importance of different regions (for e.g., see [6, 8]). Section 2.2 develops a mathematical model for the dynamics of the robot, and introduces a power consumption model as a mechanism to measure how much power a robot will consume as it moves in the domain executing a coverage algorithm.

2.1 Locational Costs and Lloyd's Algorithm

If the area of interest is $D \subset \mathbb{R}^2$, then the coverage problem involves placing N robots in D. A natural choice for dividing the domain is to let Robot i be in charge of the points of D that are closest to it. Denoting $x_i \in D, i = 1, \ldots, N$ as the positions of the robots, we then divide the area into subsets V_i, called Voronoi cells [16], which are defined as

$$V_i(x) = \{q \in D \mid \|q - x_i\| \leq \|q - x_j\|, i \neq j\}. \tag{2.1}$$

This partition of D is known as a Voronoi tessellation, denoted as V_i. Since a large class of sensors deteriorate with a rate proportional to the square of the distance from them [17], the

measure of how well a point $q \in D$ is covered by robot i at a position $x_i \in D$ is computed using the square of the Euclidean norm. The cost of coverage associated with a group of agents covering a given area can be described as,

$$C_{cov}(x,t) = \sum_{i=1}^{N} \int_{V_i(x)} \|q - x_i(t)\|^2 dq. \tag{2.2}$$

In [18], [19] it was shown that the gradient of the cost of coverage, described in Equation 2.2, as a function of the position of agent i is given as

$$\frac{\partial C_{cov}}{\partial x_i} = \int_{V_i} -2\,(q - x_i)^T dq. \tag{2.3}$$

Further, the mass and center of mass for the i^{th} Voronoi cell, V_i, can be described as

$$m_i(x) = \int_{V_i(x)} dq, \text{ and } \quad c_i(x) = \frac{\int_{V_i} q\, dq}{m_i}. \tag{2.4}$$

With these, the partial derivative in Equation 2.2 can be rewritten as

$$\frac{\partial C_{cov}}{\partial x_i} = 2\, m_i (x_i - c_i)^T. \tag{2.5}$$

By looking at Equation 2.5, we can see that a gradient direction would be $x_i - c_i$, and therefore a gradient descent motion for a network of agents would be

$$\dot{x}_i = -k\,(x_i - c_i). \tag{2.6}$$

This is known as Lloyd's algorithm, and will be referenced multiple times in this thesis as a mechanism to perform coverage control with a team of robots. Figure 2.1 provides an example of a Voronoi diagram and the theory presented in Section 2.1 for reference. In Figure 2.1 the centroid, represented as a single red cross, is at the weighted center of its cell, c_i. The robot, represented as a single blue circle, is at a given position x_i. A single robot is drawn in a particular cell to illustrate the algorithm in use, and is shown to be moving in

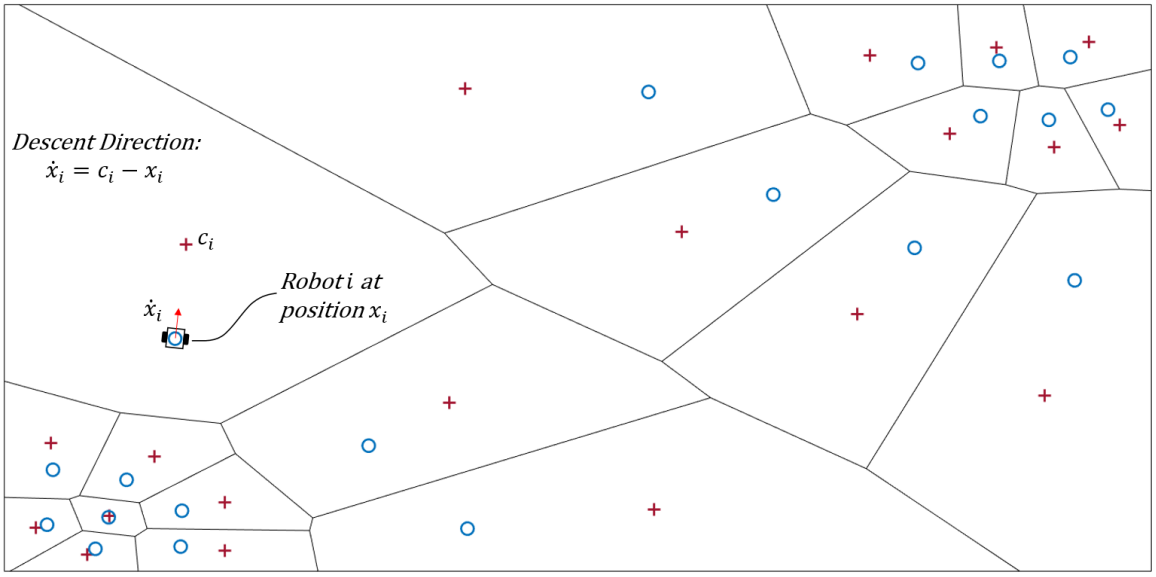

Figure 2.1: A Voronoi diagram generated in matlab that has been annotated to visually display the motion of Lloyd's algorithm.

the direction of the centroid.

2.2 Robot Dynamics and Energy Model

We now introduce the dynamics and energy model that will be used to evaluate the energy consumed by each robot as a function of its motion in the region. Differential drive robots are used as the agents in the coverage problem whose state is comprised of position and orientation. We will let the 2-D position of robot i be described as (q_{i1}, q_{i2}), with the direction of motion being q_{i3} having a linear velocity υ_i, and angular velocity ω_i. From this, the kinematics become

$$\begin{aligned} \dot{q}_{i1} &= \upsilon_i \, cos \, q_{i3} \\ \dot{q}_{i2} &= \upsilon_i \, sin \, q_{i3} \, . \\ \dot{q}_{i3} &= \omega_i \end{aligned} \tag{2.7}$$

Because Lloyd's algorithm in continuous time involves evaluating the centroid, moving towards it for a period of time, and then repeating this process, we can assume that robots

always move in straight lines (i.e. $\omega_i = 0$). We consider the energy dissipated in reorienting to face the centroid negligible in comparison to the energy consumed in the linear motion. Therefore, we turn our attention to analyzing the energy consumed in the straight line, 1-D motion. The distance a robot has moved along its straight line path is denoted as x_{i1}, and $\dot{x}_{i1} = \upsilon_i$ is robot i's velocity. In order to look at battery usage it is required that we not only look at the kinematics of the robot, but also the dynamics. We let $x_{i2} = \upsilon_i$, with dynamics $x_{i2} = u_i$, where u_i is the velocity input into the robot allowing us to build off of the simplicity of single integrator dynamics. Lastly, we let x_{i3} be the available battery level of the robot, and use a simple model of battery consumption based off of the velocity of the robot. This gives us the dynamics

$$\begin{aligned} \dot{x}_{i1} &= x_{i2} \\ x_{i2} &= u_i \qquad\quad , \\ \dot{x}_{i3} &= -\alpha_i \|u_i\|^2 \end{aligned} \tag{2.8}$$

where $\alpha_i > 0$ is a constant that determines how strongly velocity affects the battery level of each robot. Observing the equation for $\dot{x}_{i3}$, we can see that it will always be either negative or zero in the case of no motion (i.e. $\dot{x}_{i3} \leq 0, \, \forall t$).

With these dynamics, we must find a suitable model for how energy is depleted through motion of the robots. Since the instantaneous power is modeled as the euclidean norm of the velocity, the energy consumed can be found through integrating the power over time. Integrating the velocity to obtain distance results in the following equation for the energy consumed by a robot traveling from position x_{i10} to x_{i1T}, in time T, as

$$x_{i3T} - x_{i30} = -\alpha_i \| \int_0^T \dot{x}_{i1} dt \|^2 = -\alpha_i \|x_{i1}(T) - x_{i1}(0)\|^2. \tag{2.9}$$

This value is always non-positive, indicating that the energy stored in the robot is non-increasing, and the variable α indicates how strongly velocity affects the change in battery

level. The model in Equation 2.9 is sufficient since a robot's movement is nearly linear over small time horizons (T). Equation 2.9 and Equation 2.2 provide the tools to define a cost function which accounts for both coverage and energy consumption. The next chapter utilizes these metrics to perform a trade-off between the quality of coverage and the energy consumption of the robots.

CHAPTER 3

ALGORITHMS

As was stated in Chapter 1, we aim to find an algorithm that reduces coverage cost, such as Lloyd's algorithm, while balancing the energy consumed due to motion. We represent the dynamics of each robot as a hybrid dynamical system able to operate in two distinct modes. The two modes for robot i, at position x_i are

$$\dot{x}_i = \begin{cases} k\,(c_i - x_i), \\ \\ 0, \end{cases} \quad , \tag{3.1}$$

where k is a positive gain chosen to influence the speed of the robot, and c_i is the centroid of the i^{th} Voronoi cell. In Section 2.2 we defined x_{i1} and x_{i2} as position and velocity respectively. Throughout Chapter 3 we denote the position and velocity as x_i and $\dot{x}_i$ respectively, to keep notation clear and understandable. The battery level, x_{i3}, will use the notation defined in Chapter 2. Chapter 3 focuses on how to choose between the to modes defined in Equation 3.1 to conserve energy for persistent monitoring of a region.

3.1 Global Cost

Before developing a solution, we define two global metrics by which we will measure the cost of coverage and the cost of energy consumption as a function of time. The coverage metric is $C_{cov}(x,t)$, as described in Equation 2.2 across all N robots. It is desired that, as with Lloyd's, when an algorithm is implemented on a team of robots it will cause this coverage metric to decrease over time (i.e. $C_{cov}(x(t)) \leq C_{cov}(x(\tau)), \quad \forall\, t > \tau$).

We define an energy metric as

$$C_{batt}(t) = \|x_3(t_0) - x_3(t)\|, \tag{3.2}$$

where $x_3(t)$ is a column vector of every robots individual battery level defined as

$$x_3(t) \in R^N = \begin{bmatrix} x_{31}(t) \\ x_{32}(t) \\ \vdots \\ x_{3N}(t) \end{bmatrix}.$$

$C_{batt}(t)$, in Equation 3.2, is the norm over all individual robot's battery levels, and is a single scalar measure of the cumulative amount of energy consumed as a function of time. The purpose for establishing these two global cost metrics is to structure a framework across all individual robots that measures the global cost of their movements. Since $\dot{x}_{i3}(t) \leq 0, \forall\, t$, we know that $C_{batt}(t)$ will always be either increasing or constant with time just as $C_{cov}(x,t)$ will always be decreasing or constant.

We wish to reduce the cost of coverage, $C_{cov}(x,t)$, by moving towards the centroid while also limiting the amount of energy consumed, $C_{batt}(t)$, by not moving towards the centroid. This trade-off can be captured in a combination of the above two costs to give a combined overall global cost which incorporates coverage and energy as

$$C(t) = C_{Cov}(x,t) + C_{batt}(t). \tag{3.3}$$

The problem addressed in this thesis can now be defined as finding an algorithm that moves agents in a manner that reduces the global cost function in Equation 3.3. Since we wish to keep all methods and algorithms developed in this thesis decentralized, we will not be simply using minimization techniques directly on this global cost function since it encompasses all agents. We instead focus on the logic by which an individual robot would switch

between moving towards the centroid and remaining stationary. Approaching the problem from the perspective of each individual robot ensures that the sharing of information between agents with their neighbors is not required.

3.2 Approach: Threshold-Based Switching

The first solution for how to switch between the binary modes of operation, shown in Equation 3.1, modifies Lloyd's algorithm by making a decision between whether or not an individual robot finds it "worth it" to move. This is done by repeatedly comparing two different metrics at a known evaluation instant. The metrics for coverage and energy respectively are

$$Gain_{cov,i} = \int_{V_i(x)} \|q - c_i\|^2 dq - \int_{V_i(x)} \|q - x_i(t)\|^2 dq, \tag{3.4}$$

and

$$Gain_{batt,i} = -\alpha_i \|c_i - x_i(t)\|^2. \tag{3.5}$$

Equation 3.4 is taken from Equation 2.2 for a single agent, and is the difference between the coverage cost of being positioned at the centroid, and the current position of the robot. Equation 3.5 is taken from Equation 2.9 in Chapter 2 and is proportional to the square of the distance traveled. The gain in coverage can be viewed as the amount that $C_{Cov}(x,t)$ would decrease if agents were instantaneously positioned at the centroid while holding the Voronoi cell dimensions constant. The gain of energy can be viewed as proportional to the amount that $C_{batt}(t)$ would increase by if the robot moved to the centroid. The algorithm's decision to switch between modes is made by comparing, at the end of each time interval

(ΔT), the gain in coverage with the gain of energy according to

$$\dot{x}_i = \begin{cases} k\,(c_i - x_i), & \text{if } (1-\gamma)\,|Gain_{cov,i}| > \gamma\,|Gain_{batt,i}| \\ 0, & \text{otherwise} \end{cases}. \tag{3.6}$$

Equation 3.6 compares the absolute value of the improvement in coverage with the absolute value of the energy cost weighted by the relative importance of both by $\gamma \in [0, 1]$. This will result in movement over the next time horizon if the coverage improvement outweighs the energy cost, and conversely results in the robot remaining stationary if the energy cost outweighs the coverage improvement. The time horizon (ΔT) should be chosen according to the rate at which the Voronoi cells change since they are assumed to be stationary in the coverage gain equation. Computational time will rise as the time horizon is made smaller. Therefore, it is left to the designer to determine what is adequate based off the rate at which the Voronoi boundaries are changing and the computational resources at hand. This logic for switching between modes requires only a robot's current position and centroid, ensuring the algorithm remains decentralized as described in Chapter 1. In the next section we take advantage of the hybrid dynamical representation of the robot dynamics to formulate a more structured way to switch between the two modes using ideas from switch-time optimization [15].

3.3 Approach: Switch-Time Optimization

In this section we present a new method of solving the power-aware coverage control problem which is entirely independent from the threshold-based switching presented in Section 3.2. Currently at the beginning of every time interval (ΔT), the threshold-based switching algorithm determines if it is more advantageous for each robot to move or to remain stationary. From that comes a question: if a robot is moving, and on the next horizon cycle it decides it is not worth it to move, when should it have truly stopped moving? Switch-time optimization such as in [20], can be used to evaluate when to start or stop motion for each

individual robot over a finite time horizon as defined below

$$\min_{\tau_i} \int \beta E_i + (1-\beta) C_i \, dt$$

s.t.

$$\dot{x}_i = \begin{cases} k\left(c_i - x_i(t)\right), & t \leq \tau_i \\ 0, & t > \tau_i \end{cases}.$$

Where E_i is some measure of the energy consumed of robot i and C_i is some measure of the cost of coverage for robot i. Additionally the dynamic modes can be reversed for the case where a robot is not moving and then begins to drive towards the centroid.

3.3.1 Switch-Time Formulation

Switch-time optimization is a method by which to choose when to switch between different modes of dynamics so as to minimize a given cost function over time. It is desired that the integral with respect to time of E_i and C_i will result in a measure of the change in cost of coverage combined with a change in the amount of energy stored. This thought can be expressed mathematically as the difference in energy joined with the difference in coverage over the time horizon, T, as follows

$$\min_{\tau_i} (1-\beta)(C_i(T) - C_i(0)) + \beta(E_i(T) - E_i(0)). \tag{3.7}$$

The change in coverage cost can be assumed to always be negative since driving to the centroid will decrease the cost of coverage. Since our goal is to minimize the cost of coverage we want to minimize this change, making it as negative as possible. Further, the change in energy is assumed to always be negative or zero since the derivative of energy, power, is always negative as described in Equation 2.8. Since the goal is to minimize the

amount of energy consumed, we seek to minimize the difference between the initial battery level and the final battery level (i.e. we seek to minimize $E_i(0) - E_i(T)$). Therefore, we negate the energy term, E_i, in Equation 3.7. The resulting new switch-time optimization cost equation is

$$\min_{\tau_i} \int (1-\beta)C_i - \beta E_i \, dt = J(\tau_i),$$

where E_i is not only some measure of the energy consumed, but the derivative of energy with respect to time. Similarly, C_i is not only a measure of the cost of coverage, but the derivative of how the cost is changing with respect to time. This formulation ensures that when evaluating the integral over time, the intuition described in Equation 3.7 is achieved.

3.3.2 Deriving Cost Function

Since the derivative of energy with respect to time is simply the instantaneous power consumed, E_i becomes $\dot{x}_{i3}$ as was described in Equation 2.8 as

$$E_i = -\alpha \|\dot{x}_i\|^2. \tag{3.8}$$

Determining the derivative of the cost of coverage for a particular cell consists of removing the summation over all robots from Equation 2.2, and taking the derivative of it with respect to time. Therefore, C_i is defined as

$$C_i = \frac{d}{dt} \int_{V_i(x)} ||q - x_i(t)||^2 dq. \tag{3.9}$$

Calculating the time derivative in Equation 3.9 is as follows

$$\frac{d}{dt} \int_{V_i(x)} \|q - x_i(t)\|^2 dq \equiv \int_{V_i(x)} \frac{d}{dt} \|q - x_i(t)\|^2 dq.$$

Remembering the definition of a norm and inner product while ignoring the spacial integral,

we rewrite this as

$$\frac{d}{dt}\langle q - x_i, q - x_i\rangle \equiv \frac{d}{dt}(\langle q, q\rangle - 2\langle q, x_i(t)\rangle + \langle x_i(t), x_i(t)\rangle).$$

Since q does not depend on time, we have two terms to differentiate as

$$\frac{d}{dt} - 2\langle q, x_i(t)\rangle = -2\, q^T \dot{x}_i(t),$$

and,

$$\frac{d}{dt}\langle x_i(t), x_i(t)\rangle = 2\, x_i(t)^T \dot{x}_i(t).$$

Therefore, C_i can be found by combining the above to obtain the time derivative of the locational cost for a single agent in Equation 2.2 as

$$C_i = \int_{V_i(x)} 2(x_i(t) - q)^T \dot{x}_i(t) dq.$$

This allows for the energy-aware problem to be formally defined as a switch time optimization problem as

$$\begin{aligned} &\min_{\tau_i} \int_0^T \beta\alpha \|\dot{x}_i(t)\|^2 + (1-\beta) \int_{V_i(x)} 2(x_i(t) - q)^T \dot{x}_i(t)\, dq\, dt = J(\tau_i), \\ &\qquad s.t. \\ &\qquad\qquad \dot{x}_i = \begin{cases} k\,(c_i - x_i(t)), & t \leq \tau_i \\ 0, & t > \tau_i \end{cases}. \end{aligned} \tag{3.10}$$

Where in Equation 3.10, $\beta \in [0, 1]$ is a constant indicating the importance of energy as compared to coverage; α is a parameter that dictates the effect of velocity on the drain rate of the battery; x_i and $\dot{x}_i$ are the position and velocity of robot i respectively; T is the time horizon, or next evaluation instant ΔT seconds away; c_i is the centroid of the i^{th} Voronoi

cell; k is the gain of the dynamics indicating how quickly the robots move towards the centroid; and τ_i is the switch time of robot i.

3.3.3 Solution and Algorithm

The structure of Equation 3.10 is of the form of the switch time optimization problem presented in [20] and is given for reference as

$$\begin{aligned} &\min_{\tau} \int_0^T L(x(t))\, dt = J(\tau), \\ &s.t. \\ &\dot{x} = \begin{cases} f_1(x), & t \leq \tau \\ f_2(x), & t > \tau \end{cases}, \\ &\text{and} \\ &x(0) = x_0. \end{aligned} \tag{3.11}$$

For the formulation in Equation 3.10, $L(x(t))$ is

$$L(x(t)) = \beta \|\dot{x}_i(t)\|^2 + (1-\beta) \int_{V_i(x)} 2(x_i(t) - q)^T \dot{x}_i(t)\, dq,$$

and $f_1(x)$ and $f_2(x)$ in $\dot{x}$ are

$$\dot{x}_i = \begin{cases} f_1(x) = k\,(c_i - x_i(t)), & t \leq \tau_i \\ f_2(x) = 0, & t > \tau_i \end{cases}.$$

The switch-time problem in Equation 3.11 has a solution obtained through calculus of variations as in [20], and when applied to our formulation in Equation 3.10 is

$$\begin{aligned}
&\frac{\partial \tilde{J}}{\partial \tau_i} = \lambda_i^T(\tau_i)[f_{1i}(x_i(\tau_i)) - f_{2i}(x_i(\tau_i))] \stackrel{!}{=} 0 \\
&\dot{\lambda}_i(t) = -\frac{\partial L^T}{\partial x_i} - \frac{\partial f_{2i}^T}{\partial x_i}\lambda_i, \quad on\ [\tau_i, T] \qquad . \\
&\lambda_i(T) = 0
\end{aligned} \tag{3.12}$$

For the formulation of the switch time problem in Equation 3.10, we can see that when $f_{2i}(t) = 0 \quad \forall t$, the problem is drastically simplified. First, the right term of the derivative of the co-state, $\dot{\lambda}_i(t)$, becomes zero. Second, the state of the system, $x_i(t)$, will not change after time τ_i since the dynamics $\dot{x}_i(t)$ are equal to 0 on the interval $[\tau_i, T]$.

We must first solve for $\frac{\partial L}{\partial x_i}$, then negate and transpose it to find $\dot{\lambda}_i(t)$. To obtain $\frac{\partial L}{\partial x_i}$ we must take the spatial derivative of the cost function as shown below

$$\frac{\partial L}{\partial x_i} = \beta\,\alpha\,\frac{\partial}{\partial x_i}\left\|\dot{x}_i(t)\right\|^2 + (1-\beta)\int_{V_i(x)} \frac{\partial}{\partial x_i} 2(x_i - q)^T \dot{x}_i(t)\,dq.$$

Remembering that $\frac{\partial}{\partial x}\left\|x\right\|^2 = 2\,x^T$, and that $\dot{x}_i(t) = k\,(c_i - x_i(t))$ we obtain

$$\frac{\partial L}{\partial x_i} = 2\beta\,\alpha\,k^2(x_i(t) - c_i)^T + (1-\beta)\int_{V_i(x)} 2c_i^T + 2q^T - 4x_i^T(t)\,dq. \tag{3.13}$$

Integrals with respect to the variable q are defined in Equation 2.4, and with their definitions Equation 3.13 can be simplified to

$$\frac{\partial L}{\partial x_i} = 2\beta\,\alpha\,k^2(x_i(t) - c_i)^T + (1-\beta)\,((2\,c_i^T - 4\,x_i^T(t))\,m_i + 2\,c_i^T\,m_i). \tag{3.14}$$

Where m_i and c_i is the mass and centroid of the Voronoi cell respectively, defined in Equation 2.4. Since x_i is a 2 x 1 column vector, $\frac{\partial L}{\partial x_i}$ is a 1 x 2 row vector. Transposing and

negating to find $\dot{\lambda}_i(t)$ becomes

$$\dot{\lambda}_i(t) = 2\beta\,\alpha\,k^2(c_i - x_i(t)) + (1-\beta)\,((4\,x_i(t) - 2\,c_i)\,m_i - 2\,c_i\,m_i) \tag{3.15}$$

This is a 2 x 1 column vector which is

$$\dot{\lambda}_i(t) = \begin{bmatrix} \dot{\lambda}_{i,1}(t) \\ \dot{\lambda}_{i,2}(t) \end{bmatrix} = \begin{bmatrix} 2\beta\,\alpha\,k^2(c_{i,1} - x_{i,1}) + (1-\beta)\,((4\,x_{i,1} - 2\,c_{i,1})\,m_i - 2\,c_{i,1}\,m_i) \\ 2\beta\,\alpha\,k^2(c_{i,2} - x_{i,2}) + (1-\beta)\,((4\,x_{i,2} - 2\,c_{i,2})\,m_i - 2\,c_{i,2}\,m_i) \end{bmatrix}. \tag{3.16}$$

Since the state of the system, $x_i(t)$, is not changing with time, $\dot{\lambda}_i(t)$ is constant on $[\tau_i, T]$. A numerical algorithm, similar to [20], can be developed to find the optimal switch time. The pseudo code is described in Algorithm 1.

Algorithm 1 Switch-time gradient descent

1: `k = 0`
2: `Guess` τ_{i0}
3: `repeat:`
4: `Simulate` x_i `forward in time from` x_{i0}
5: `Simulate` λ_i `backwards in time from` $\lambda_i(T) = 0$ `to obtain` $\lambda_i(\tau_{ik})$
6: `Update` τ_{ik+1} `as follows:` $\tau_{ik+1} = \tau_{ik} - \gamma\,\lambda_i^T(\tau_{ik})\,[f_{1i}(x_i(\tau_{ik})) - f_{2i}(x_i(\tau_{ik}))]$

Lastly, the dynamics must be reversed for the case in which a robot is not moving and then switches to begin to drive towards the centroid. The dynamics for this case are

$$\dot{x}_i = \begin{cases} f_{1i}(x_i) = 0, & t \le \tau_i \\ f_{2i}(x_i) = k\,(c_i - x_i(t)), & t > \tau_i \end{cases}.$$

This problem formulation still has the same known solution obtained through calculus of

variations in [20], and defined in Equation 3.11. We have already obtained $-\frac{\partial L}{\partial x_i}^T$ and now must solve for $\frac{\partial f_{2i}}{\partial x_i}$ in order to fully solve for $\dot{\lambda}_i(t)$.

Remembering that

$$f_{2i} = \begin{bmatrix} k\left(c_{i,1} - x_{i,1}\right) \\ k\left(c_{i,2} - x_{i,2}\right) \end{bmatrix},$$

$\frac{\partial f_{2i}}{\partial x_i}$ is

$$\frac{\partial f_{2i}}{\partial x_i} = \begin{bmatrix} \frac{\partial}{\partial x_{i,1}}(k\left(c_{i,1} - x_{i,1}\right)) & \frac{\partial}{\partial x_{i,2}}(k\left(c_{i,1} - x_{i,1}\right)) \\ \frac{\partial}{\partial x_{i,1}}(k\left(c_{i,2} - x_{i,2}\right)) & \frac{\partial}{\partial x_{i,2}}(k\left(c_{i,2} - x_{i,2}\right)) \end{bmatrix},$$

yielding

$$\frac{\partial f_{2i}}{\partial x_i} = \begin{bmatrix} -k & 0 \\ 0 & -k \end{bmatrix}.$$

Therefore, $\dot{\lambda}_i(t)$ can be written as

$$\dot{\lambda}_i(t) = \begin{bmatrix} \dot{\lambda}_{i,1}(t) \\ \dot{\lambda}_{i,2}(t) \end{bmatrix} = \begin{bmatrix} 2\beta\,\alpha\,k^2(c_{i,1} - x_{i,1}) + (1-\beta)\left((4\,x_{i,1} - 2\,c_{i,1})\,m_i - 2\,c_{i,1}\,m_i\right) \\ 2\beta\,k_p\,k^2(c_{i,2} - x_{i,2}) + (1-\beta)\left((4\,x_{i,2} - 2\,c_{i,2})\,m_i - 2\,c_{i,2}\,m_i\right) \end{bmatrix} + k \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_{i,1}(t) \\ \lambda_{i,2}(t) \end{bmatrix}. \quad (3.17)$$

Algorithm 1 can be used as a gradient descent algorithm to find the optimal switch-time for this case of dynamics as well noting that, in this case, the state x_i is not constant on the interval $[\tau_i, T]$.

In Chapter 3 we have developed two distinct approaches by which the goals of coverage

and energy conservation can be balanced through changeable parameters γ and β. The algorithms derived in this section provide a framework for balance which we claim will lead to multi-robot teams being able to operate in persistent monitoring applications for longer periods of time as compared to the canonical Lloyd's algorithm. The framework provides one with the ability to achieve energy conservation through a possible decrease in surveillance depending on the priority of each. In this thesis, we do not attempt to define an allowable decrease in surveillance or an appropriate gain in energy conservation, since these will vary depending on the application. The utility of the algorithms presented is the framework itself, to which one can fit to a specific application. In Chapter 4, the threshold-based switching and switch-time optimization algorithms are implemented in simulation.

CHAPTER 4
IMPLEMENTATION

In order to demonstrate the utility of the algorithms presented in Chapter 3, experiments were performed utilizing the algorithms in simulation. Experiments demonstrating the ability for the algorithms to balance between coverage and energy are presented in Section 4.1. Throughout this thesis the claim has been made that the ability to balance energy and coverage allows for the persistent monitoring of a region for a longer period of time than the canonical Lloyd's algorithm. To substantiate that claim, Section 4.2 presents experiments where agents, executing each algorithm, continuously monitor a dynamically changing environment and battery levels are compared.

4.1 Experiment: Balancing Coverage and Energy

To demonstrate the ability to balance coverage and energy, this section presents short experiments using the derived algorithms for static density distributions. We qualitatively tune the parameters that indicate the importance of energy as compared to coverage, γ and β, until we see a trade-off between coverage and energy. It is desired that the algorithms neither cause the robots to never move (i.e. γ and β are too high), or cause the robots to always move (i.e. γ and β are too low). We do not formally define what values of γ and β are best since this varies depending on the application and manner in which the algorithms are implemented. Therefore, we only strive to show that some measure of energy savings can be made for a trade-off in coverage cost, and ignore the degree to which the trade-off occurs.

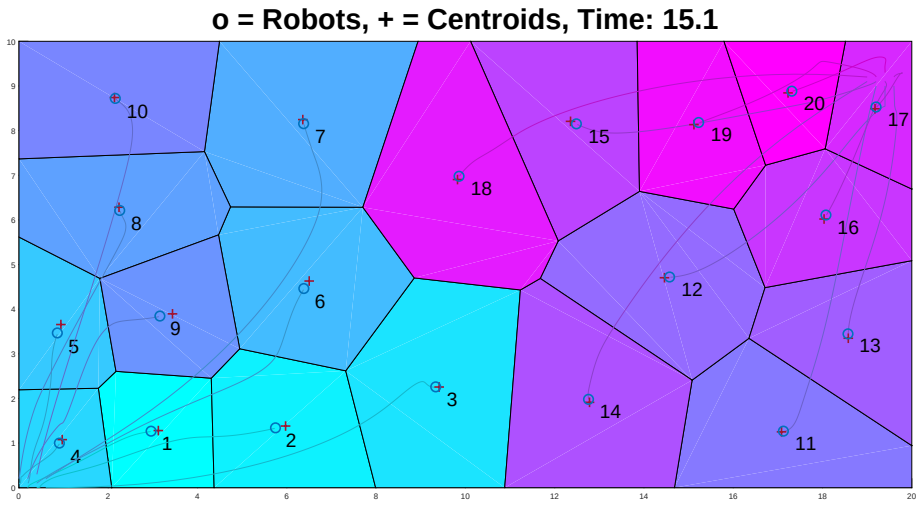

Figure 4.1: A graphic representation of simulated algorithms working in real time. The circles represent the agents which are moving towards the centroid, represented by a cross, of their individual Voronoi cell. The trailing lines from each agent depict the movement of the agent over time.

4.1.1 Simulation

Each algorithm presented above in section 2.1, 3.2, and 3.3 (i.e. Lloyds, Threshold-Based Switching, and Switch-Time Optimization) was implemented in a network of 20 simulated robots with 2-D positions in a rectangular 1.46 x 1.56 world. Given that agents in coverage problems typically originate from a single starting point due to charging location or central computing resource, the agents were clustered together in two groups (10 agents each) at the south-west and north-east corners of the area being covered. The robots were clustered in two groups so as to demonstrate a dynamic environment where the needs of coverage are changing (i.e. as agents explore the region they may find another team of agents also covering it). Algorithms were executed for 60 seconds. Each robot was given the same amount of initial battery life which was 5. Figure 4.1 shows an instant in time of the simulation graphic that was displayed while each algorithm was being executed.

The gain, k, in Equation 2.6 was chosen to be one ($k = 1$) for simplicity and did not influence the speed of the robots. The threshold-based switching algorithm required a time

cycle, ΔT, for which it would compare the two cost metrics in Equation 3.4 and Equation 3.5 in order to update the dynamics, choosing to move or not move, as shown in Equation 3.6. The time cycle was chosen to be one second ($\Delta T = 1$) because this would allow frequent re-evaluation without slowing the algorithm's speed through calculating coverage and energy gains too frequently. Similarly, to conserve computational resources, the time horizon for the switch time algorithm was set to one second ($\Delta T = 1$). The indicator for the importance of energy in the threshold-based algorithm, γ, was chosen to be 0.45 because it was a value that neither forced the robots to never move (i.e. energy importance is high enough that robots never deem it worth it to move), or always move (i.e. energy is of such low importance that robots always decide to move towards their centroid). The indicator for the importance of energy in the switch-time algorithm, β, was chosen to be 0.997 for the same reason.

It should be noted that since the left term of $Gain_{cov,i}$ in Equation 3.4 can be viewed as an approximation to the coverage cost if the agents were instantaneously positioned at the centroid, the evaluation time (ΔT) should be a small enough value that the Voronoi diagram does not change significantly over ΔT. Since this method of comparison does not model how the Voronoi cells will evolve as agents move towards their centroid, it also cannot model how the centroid itself will move. Therefore, because modeling the future evolution of the agents is computationally intensive, it is necessary to keep the time horizons small. Conversely, the value of ΔT can not be made infinitesimally small because of an increase in computational complexity.

When examining the results of the simulations, the three primary factors of interest were the global cost, average battery level, and the coverage cost across all N robots. Since the power-aware coverage problem looks to balance the energy consumed with the minimization of coverage costs, a primary indicator of a successful algorithm would be one that decreased the global cost function more than or equivalently to a baseline method (i.e. Lloyds algorithm). Figure 4.2 shows how the global cost function, as described in

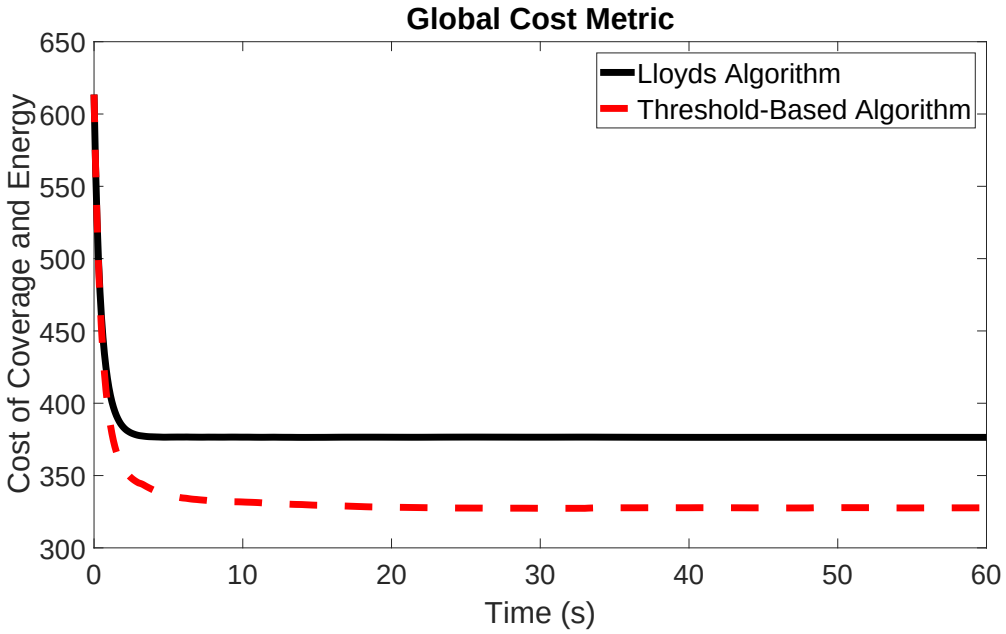

Figure 4.2: The global cost function, derived in Section 3.1, over time for Lloyd's and the Threshold-Based Switching algorithms. The initial cost is strictly the cost of coverage since no energy has been consumed at t_0.

Equation 3.3, evolved over time for Lloyd's algorithm and the threshold-based algorithm. The threshold based algorithm provides an improvement over Lloyd's algorithm as can be seen by the lower global cost value as the experiment evolved. Figure 4.3 provides the same plot for the Switch-Time Optimization algorithm. Again, the switch-time algorithm provides an improvement to Lloyd's algorithm as demonstrated by the lower global cost value as the experiment evolved.

Since the algorithms presented in this thesis look to conserve energy while providing adequate coverage, it is important to also look at how the battery levels of the agents evolved over time. Figure 4.4 and Figure 4.5 shows how the average battery levels across all robots evolved over time. In order for Figure 4.2 and Figure 4.3 to be an important indicator of a successful algorithm, it must assume that a lower global cost translates to less energy used overall. Figure 4.4 and Figure 4.5 confirms this assumption by showing that the threshold-based switching and the switch-time algorithm used less energy than Lloyd's algorithm.

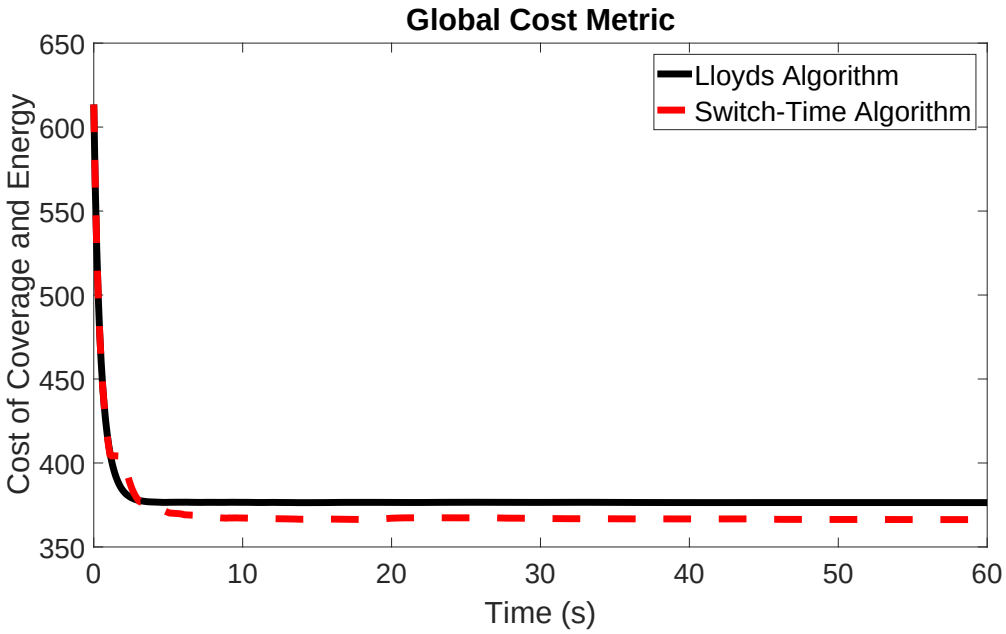

Figure 4.3: The global cost function, derived in Section 3.1, over time for Lloyd's and the Switch-Time Optimization algorithms. The initial cost is strictly the cost of coverage since no energy has been consumed at t_0.

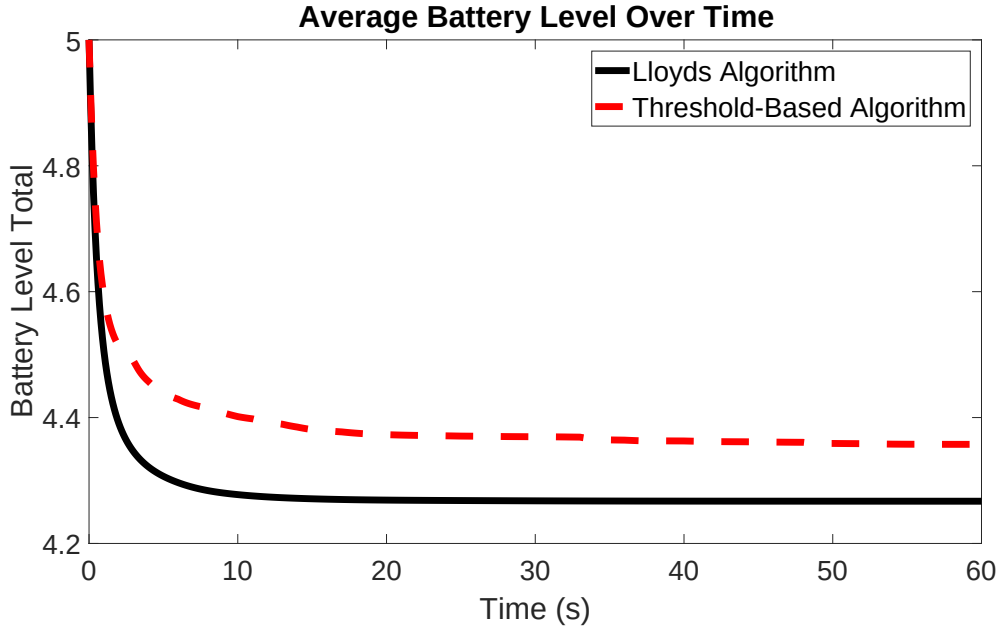

Figure 4.4: The average battery levels across all agents over time for Lloyd's and the Threshold-Based Switching algorithms. The initial value is 5 which was the initial battery value for all robots.

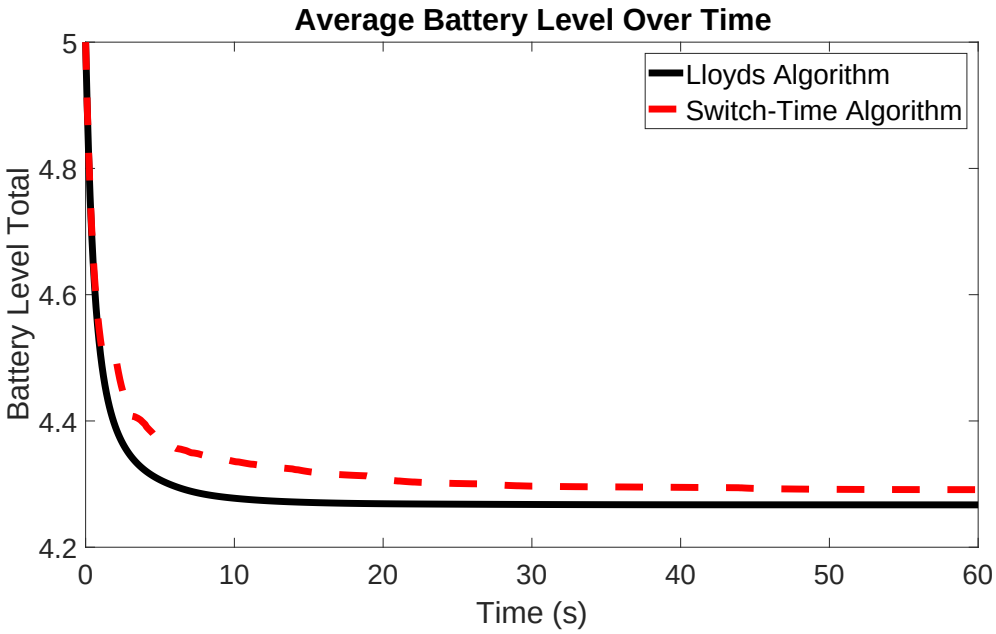

Figure 4.5: The average battery levels across all agents over time for Lloyd's and the Switch-Time Optimization algorithms. The initial value is 5 which was the initial battery value for all robots.

Lastly, we examine the compromise that was made in coverage costs. The algorithms presented in this paper are a framework to be able to quantify the balance between energy and coverage. Therefore a decrease in the amount of energy consumed resulted in an increase of the coverage cost. Figure 4.6 shows the coverage cost over time during the experiment for the threshold-based and Lloyd's algorithms. As expected, Lloyd's algorithm yielded a smaller coverage cost. Similarly, Figure 4.7 show the coverage cost throughout the experiment for the switch-time algorithm, and Lloyd's algorithm yielded a smaller coverage cost as expected.

4.1.2 Robotarium Simulator

In addition to the simulator used in Section 4.1.1, the three algorithms were implemented on a robotarium simulator such as described in [21]. The simulator models differential-drive robots with unicycle dynamics. The algorithms were implemented on a 3 x 3 world with eight agents. The robots were again clustered into two groups (4 agents each) at the

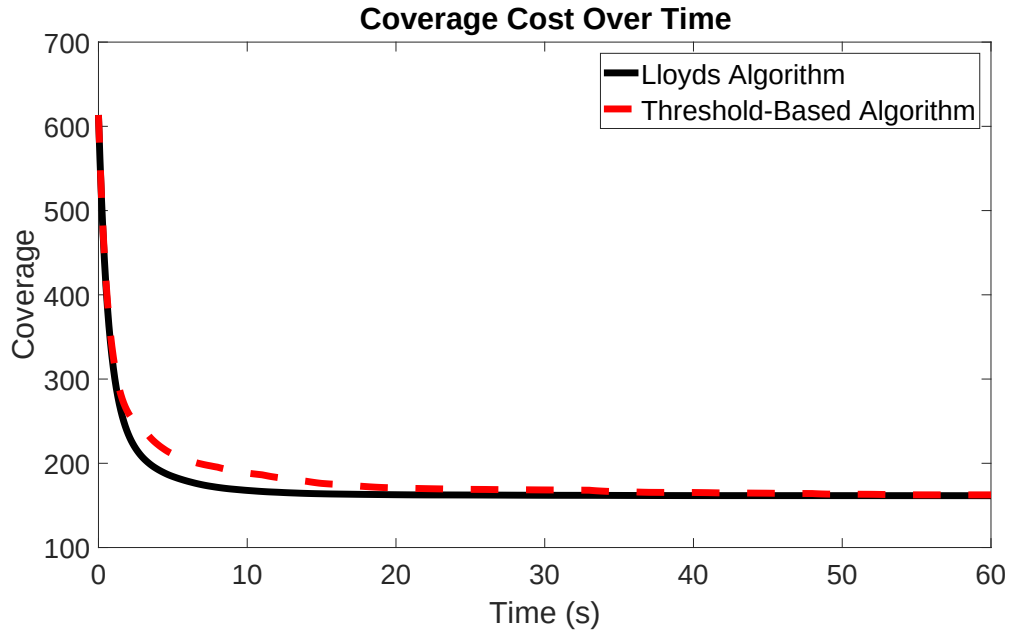

Figure 4.6: The total sum of coverage levels across all agents over time for Lloyd's and the Threshold-Based Switching algorithms.

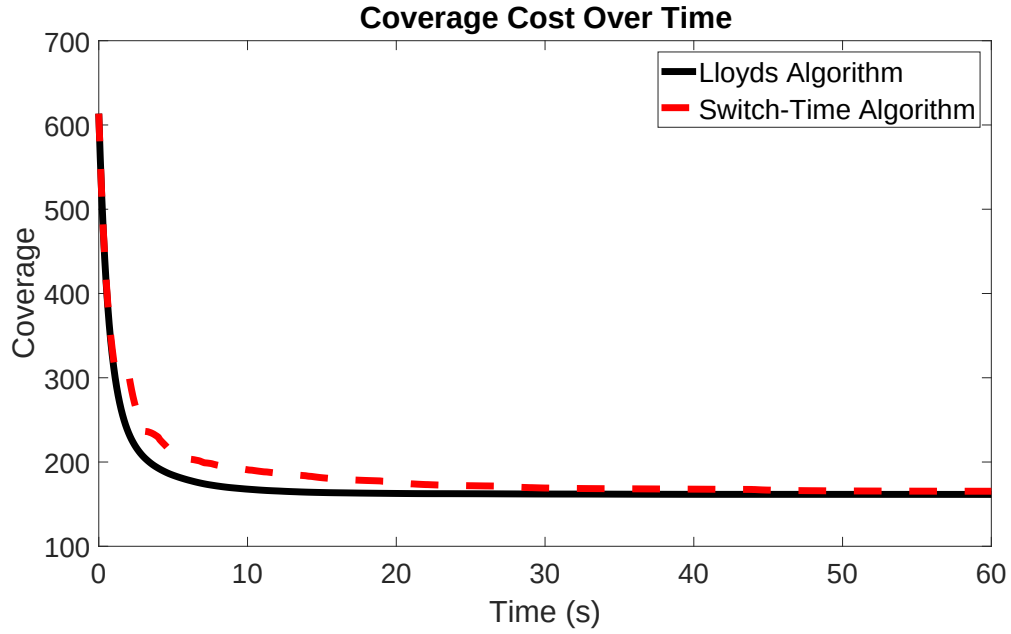

Figure 4.7: The total sum of coverage levels across all agents over time for Lloyd's and the Switch-Time Optimization algorithms.

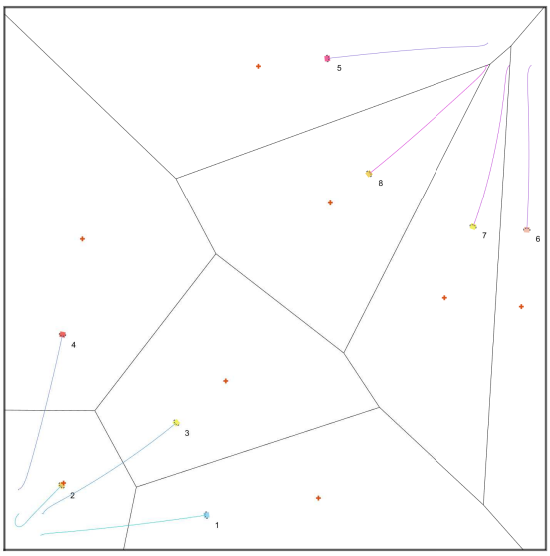

Figure 4.8: The robotarium simulator performing Lloyd's algorithm.

south-west and north-east corners of the area being covered. Algorithms were executed for 60 seconds. Each robot was given the same amount of initial battery life, which was 500. The initial battery was increased because the area being covered was larger, and as a result, more energy was consumed. The constants k, and ΔT were kept the same as in Section 4.1.1. Because the area being covered changed, different values of γ and β were used. The value indicating the importance of energy in the threshold-based algorithm, γ, was chosen to be 0.25. The value indicating the importance of energy in the switch-time algorithm, β, was chosen to be 0.997.

Figure 4.8 provides an example of the robotarium simulator running Lloyd's algorithm at a particular instant in time. Figure 4.9 shows how the global cost function, as described in Equation 3.3, evolved over time for Lloyd's algorithm and the threshold-based algorithm implemented on the robotarium simulator. Figure 4.10 provides the same plot for Lloyd's algorithm and the switch-time algorithm being implemented on the robotarium simulator. Figure 4.11 and Figure 4.12 shows how the total sum of battery levels across all robots evolved over time. Figure 4.13 shows the coverage cost over time during the experiment for the threshold-based and Lloyd's algorithms implemented on the robotarium simulator.

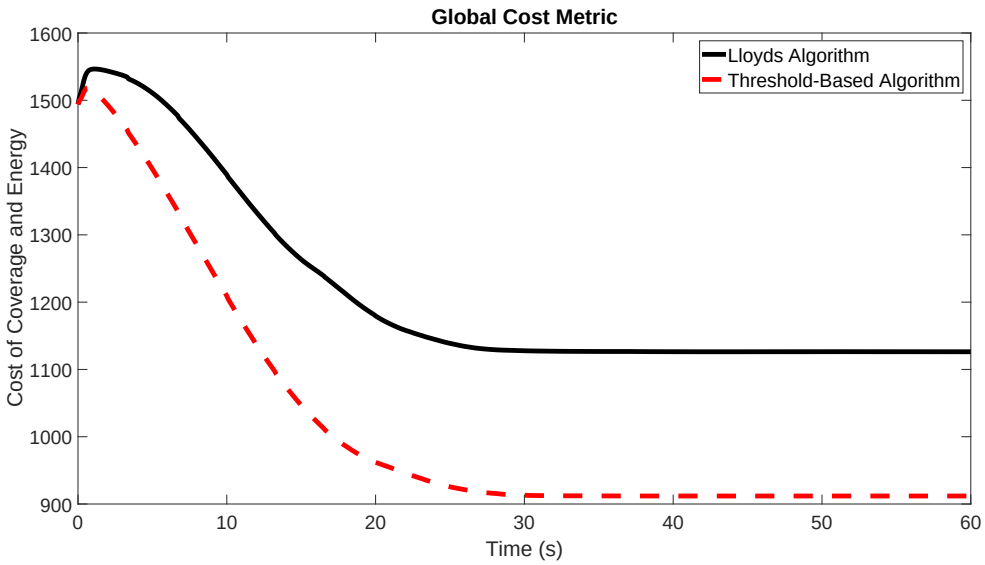

Figure 4.9: The global cost function, derived in Section 3.1, over time for Lloyd's and the Threshold-Based Switching algorithms implemented on robotarium simulator. The initial cost is strictly the cost of coverage since no energy has been consumed at t_0.

Similarly, Figure 4.14 shows the coverage cost throughout the experiment for the switch-time algorithm, and Lloyd's algorithm implemented on the robotarium simulator.

The results of running the robotarium simulator validate the results from the simulation, because similar performances of the average battery level, coverage cost, and global cost were observed. A lower global cost was achieved through the modified algorithms as compared to Lloyd's (Figures 4.9 and 4.10). A decrease in average battery level was achieved (Figures 4.11 and 4.12) through an increase in coverage cost (Figures 4.13 and 4.14). With the results of the simulations in Section 4.1.1 verified, we move on to perform a more robust experiment in Section 4.2.

4.2 Experiment: Persistent Monitoring

In order to create an environment where the true benefit of the algorithms presented in this thesis can demonstrated, we create a persistent monitoring experiment where the needs of coverage are continuously changing. In Section 4.1 an experiment was performed where the robots spread out to cover a region spatially. The persistent monitoring experiment

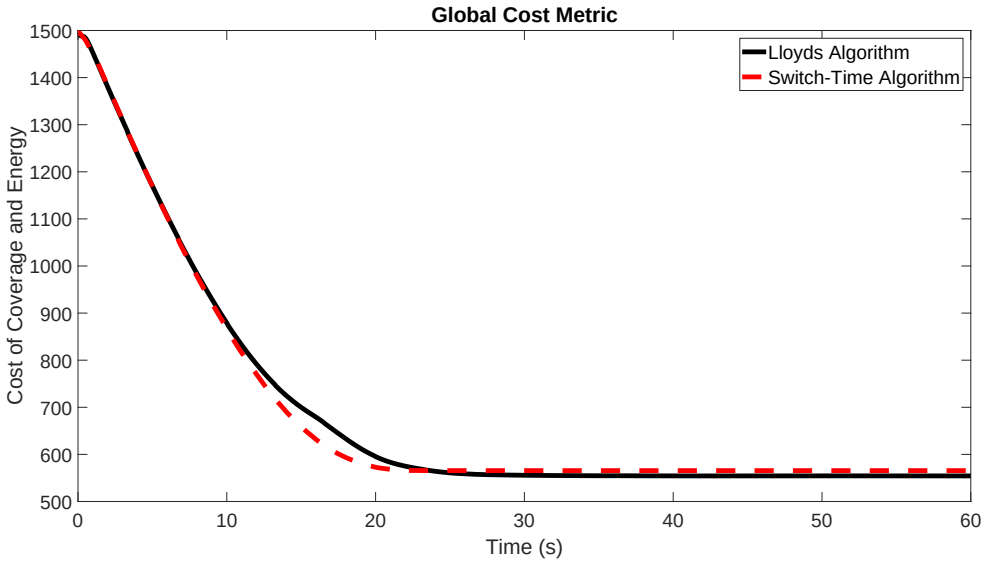

Figure 4.10: The global cost function, derived in Section 3.1, over time for Lloyd's and the Switch-Time Optimization algorithms implemented on the robotarium simulator. The initial cost is strictly the cost of coverage since no energy has been consumed at t_0.

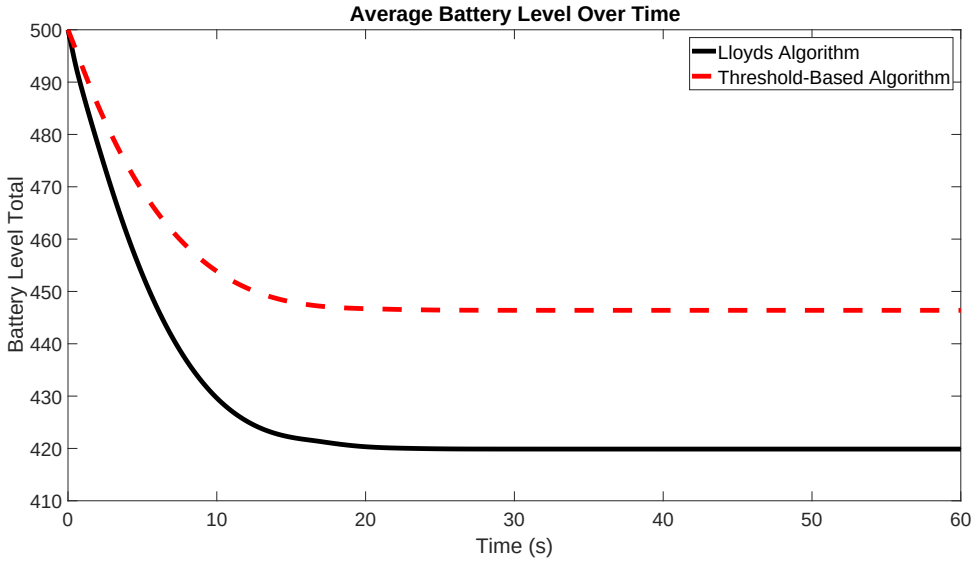

Figure 4.11: The average battery levels across all agents over time for Lloyd's and the Threshold-Based Switching algorithms implemented on the robotarium simulator. The initial value is 500 which was the initial battery value for all robots.

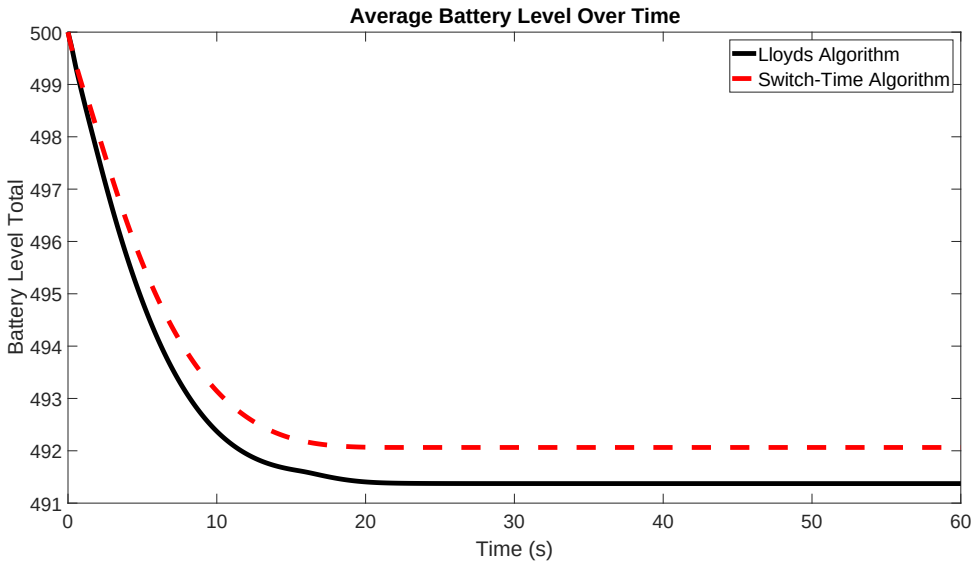

Figure 4.12: The average battery levels across all agents over time for Lloyd's and the Switch-Time Optimization algorithms implemented on the robotarium simulator. The initial value is 500 which was the initial battery value for all robots.

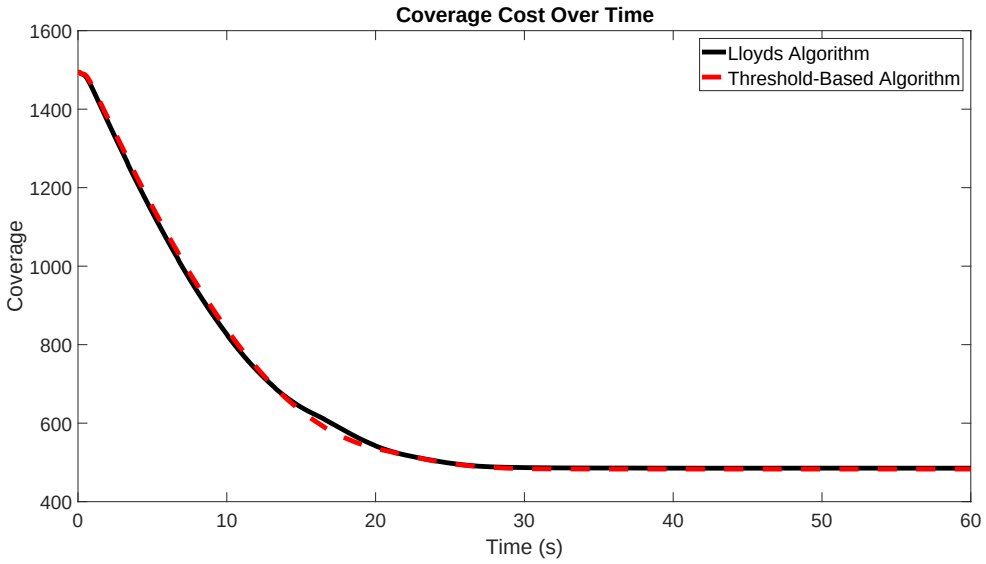

Figure 4.13: The total sum of coverage levels across all agents over time for Lloyd's and the Threshold-Based Switching algorithms implemented on the robotarium simulator.

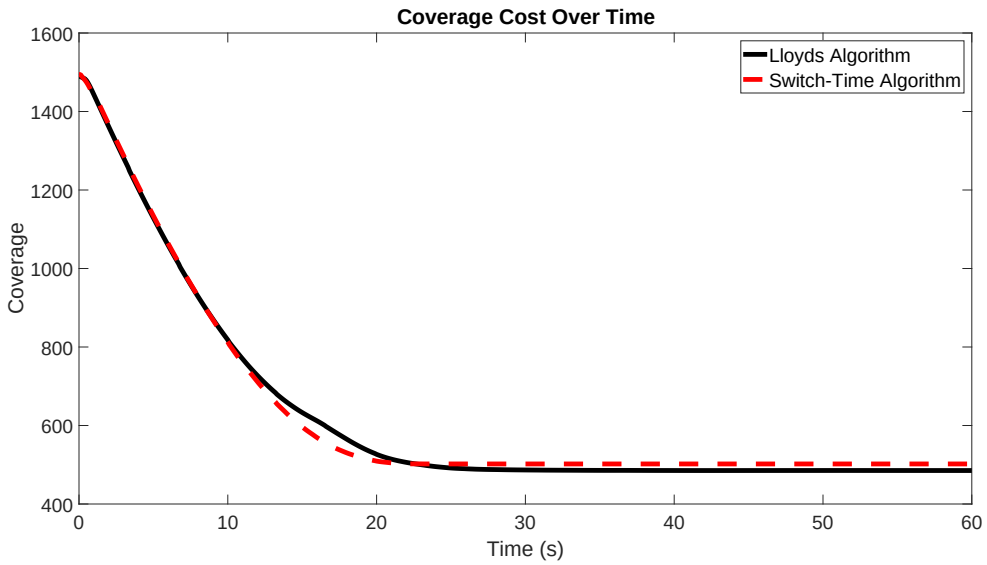

Figure 4.14: The total sum of coverage levels across all agents over time for Lloyd's and the Switch-Time Optimization algorithms implemented on the robotarium simulator.

can be thought of as this same experiment of coverage where some areas will be of higher interest than others. Further, those areas of higher interest will vary with time causing for continuous execution of the coverage algorithm.

The concept of differing regions of importance can be mathematically formulated with a density distribution function providing a weighted factor to the cost of coverage at point $q \in D$ as in [22]. This density distribution function, denoted as ϕ, is a function of both position and time. To perform this experiment we do not modify the algorithms already presented, but simply employ the density distribution function as a tool to continuously change the centroid of a cell. This requires a slight modification to the equations for the mass and centroid of a cell in Equation 2.4. The weighted mass and centroid will now be calculated as

$$m_i(x) = \int_{V_i(x)} \phi(q,t)dq \quad , \quad c_i(x) = \frac{\int_{V_i} q\phi(q,t)dq}{m_i}. \tag{4.1}$$

The algorithms presented in Chapter 3 do not need to be modified to fit a time varying density distribution provided that the rate at which the density changes is small [22]. By

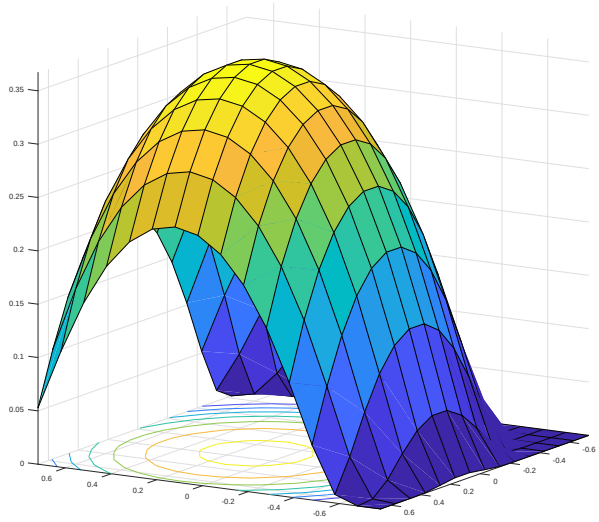

Figure 4.15: The two-dimensional Gaussian function plotted at the instant in time when it is at the far south-west corner of the region.

not modifying the threshold-based switching or switch-time optimization algorithms, we make the assumption that the density distribution is nearly constant relative to the robot. This experiment was performed in simulation similar to that described in Section 4.1.1.

The density distribution function in the following experiment was a two-dimensional Gausian density distribution $\phi(x,t) \in [0,1]$. The peak of this function shifted from the south-west corner to the north-east corner of the region. Figure 4.15 and Figure 4.16 plots the density distribution in three dimensions. Figure 4.15 displays the case where the peak of the density distribution is at the south-west corner of the region. Figure 4.16 shows the case where the peak of the density distribution is at the north-east corner of the region.

4.2.1 Simulation

To evaluate performance, we consider how much longer the multi robot team, running the improved algorithms, can sustain coverage given a finite initial battery level. Since it is difficult to construct an experiment where all robots will lose power at the same time, we choose to use the time at which the multi-robot team's average battery level has decreased by 25%. Lloyd's algorithm used 25% of its battery in 168 seconds. The binary decision

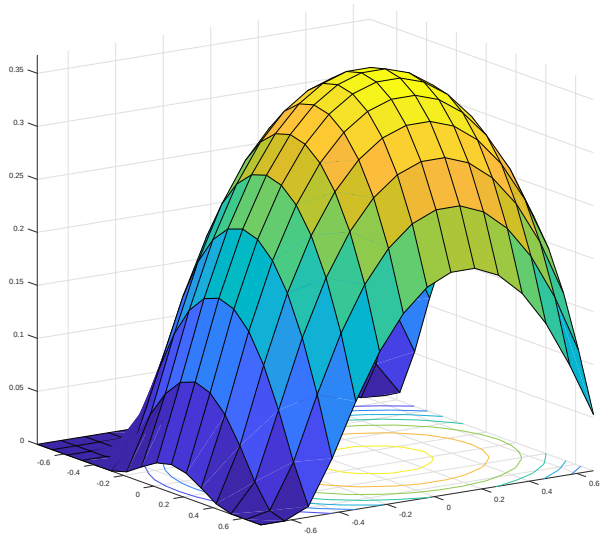

Figure 4.16: The two-dimensional Gaussian function plotted at the instant in time when it is at the far north-east corner of the region.

algorithm lasted for 100 seconds longer and used 25% of its battery power in 268 seconds. The switch-time algorithm lasted 229 seconds longer using 25% of its overall battery power in 452 seconds as compared to Lloyd's 223 seconds to reach 25% of its battery power. The reason for the difference in time that Lloyd's reached 25% of its battery power is due to the fact that the switch-time algorithm was run with a coarser mesh density to improve run time by reducing the order of computation.

In addition to looking at the amount of time that the robot-team's average battery level decreased by 25%, we can also plot the same metrics of performance that were generated in Section 4.1. In these experiments, the exact same values of γ, β, ΔT, and k, were used. The 20 robots were again placed in a $1.46x1.56$ world. The only difference was in the varied density distribution of $\phi(x,t)$, as shown in Figure 4.15 and Figure 4.16. Additionally, the initial battery level was modified slightly to a value of 3. Figure 4.17 shows how the global cost function, as described in Equation 3.3, evolved over time for Lloyd's algorithm and the threshold-based switching algorithm, implemented with a time-varying density distribution. Figure 4.18 provides the same plot for the switch-time optimization algorithm. Again, the switch-time algorithm provides an improvement to Lloyd's as demonstrated by

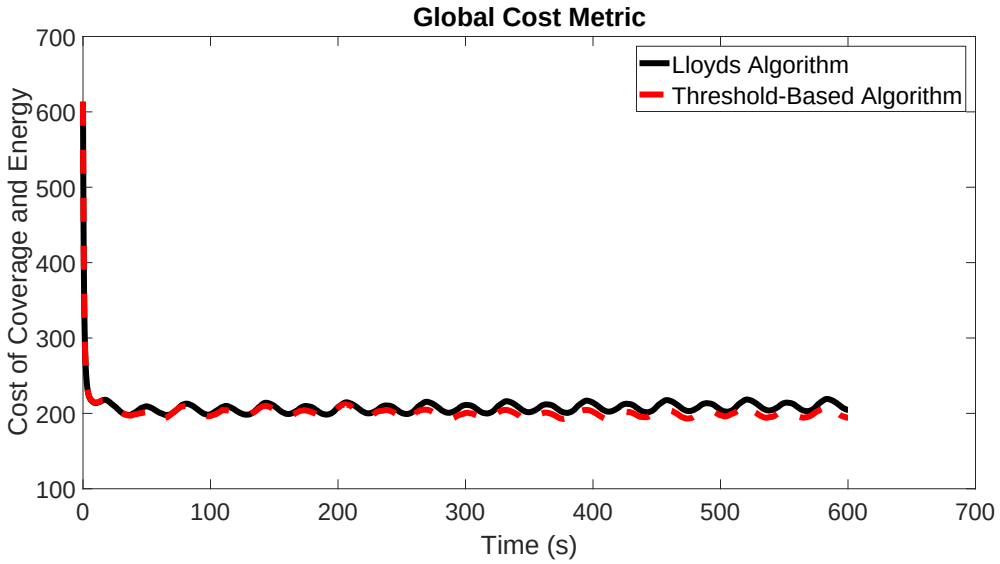

Figure 4.17: The curves of the global cost function, derived in Section 3.1, over time for Lloyd's and the Threshold-Based Switching algorithms using a time varying density distribution. The initial cost is strictly the cost of coverage since no energy has been consumed at t_0.

the lower global cost value as the experiment evolved.

Figure 4.19 and Figure 4.20 shows how the average battery level across all robots evolved over time, with a time varying density distribution, for the threshold-based and switch-time algorithms respectively.

Figure 4.21 shows the coverage cost over time during the experiment for the threshold-based and Lloyd's algorithms using a time varying density distribution. As expected, Lloyd's algorithm yielded a smaller coverage cost. Similarly, Figure 4.22 shows the coverage cost throughout the experiment for the switch-time algorithm using a time varying density distribution, and Lloyd's algorithm yielded a smaller coverage cost.

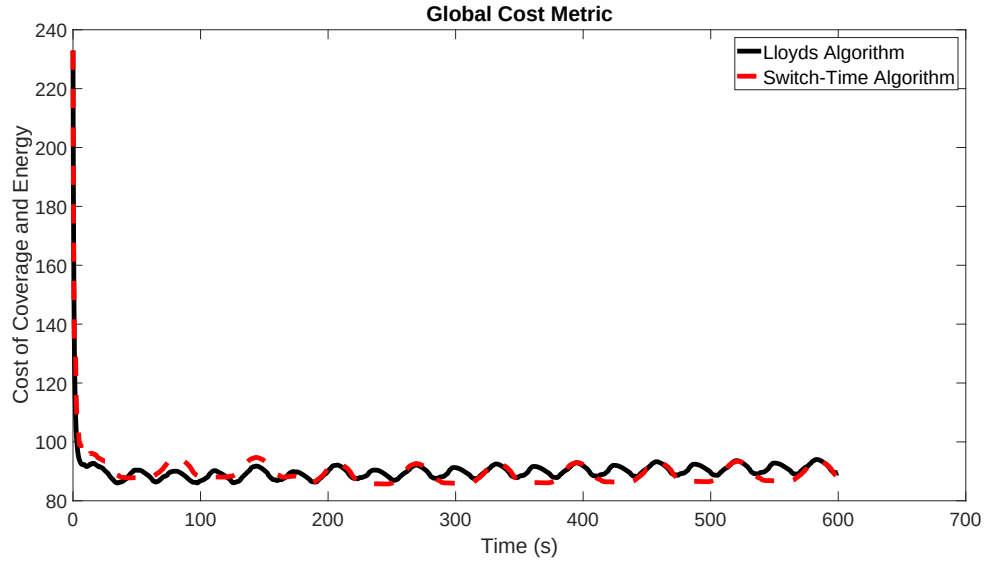

Figure 4.18: The curves of the global cost function, derived in Section 3.1, over time for Lloyd's and the Switch-Time Optimization algorithms using a time varying density distribution. The initial cost is strictly the cost of coverage since no energy has been consumed at t_0.

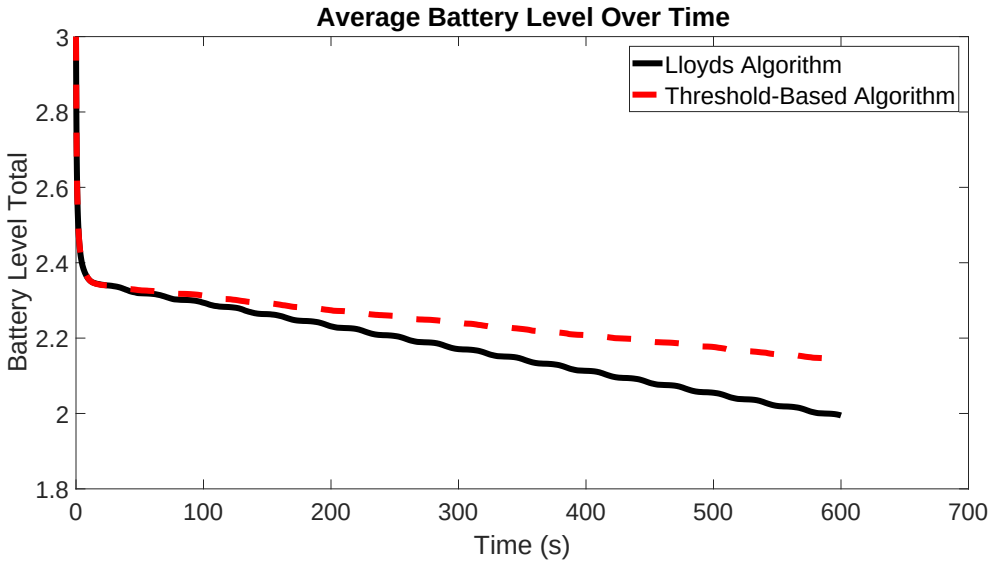

Figure 4.19: The average battery levels across all agents over time for Lloyd's and the Threshold-Based Switching algorithms using a time varying density distribution. The initial value is 3 which was the initial battery value for all robots.

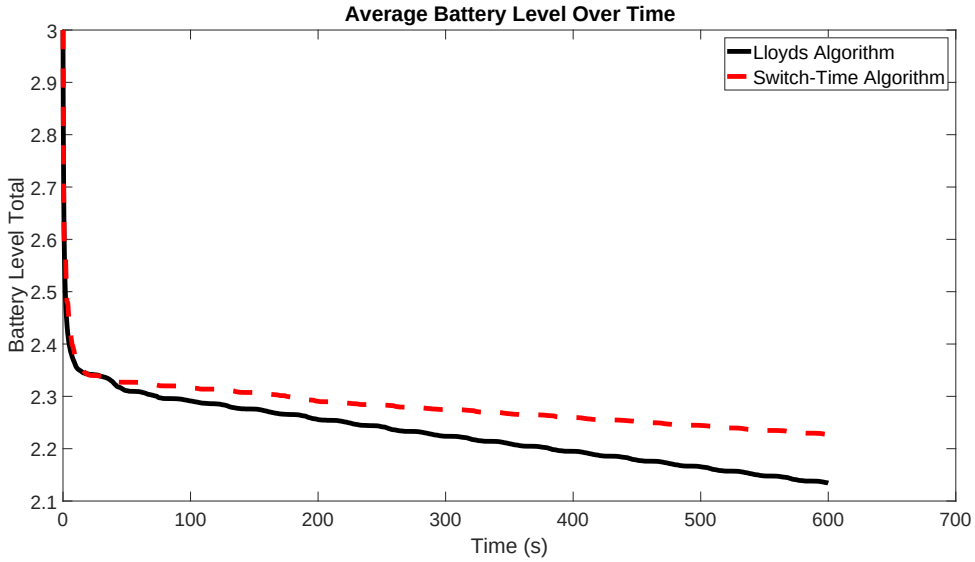

Figure 4.20: The average battery levels across all agents over time for Lloyd's and the Switch-Time Optimization algorithms using a time varying density distribution. The initial value is 3 which was the initial battery value for all robots.

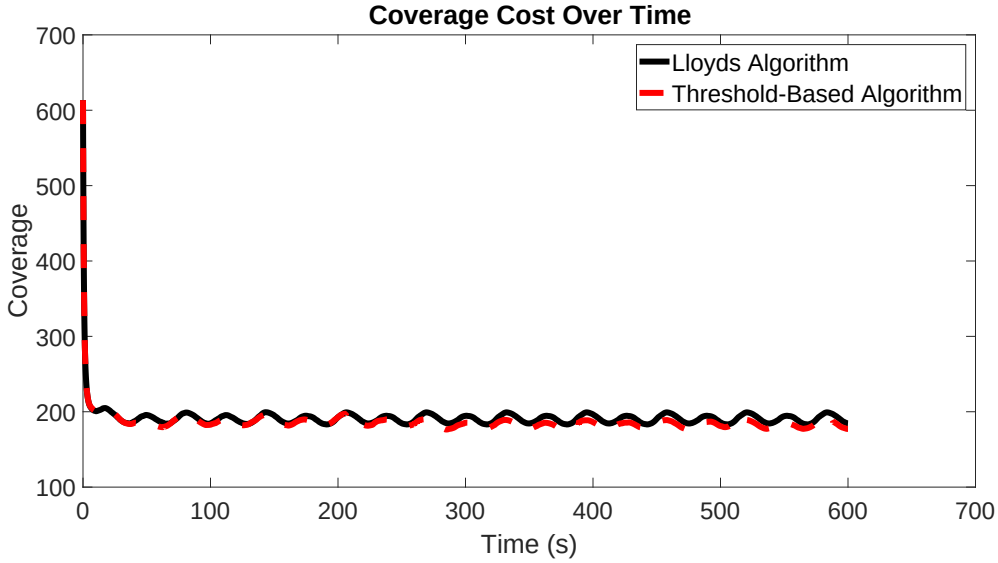

Figure 4.21: The total sum of coverage levels across all agents over time for Lloyd's and the Threshold-Based Switching algorithms using a time varying density distribution.

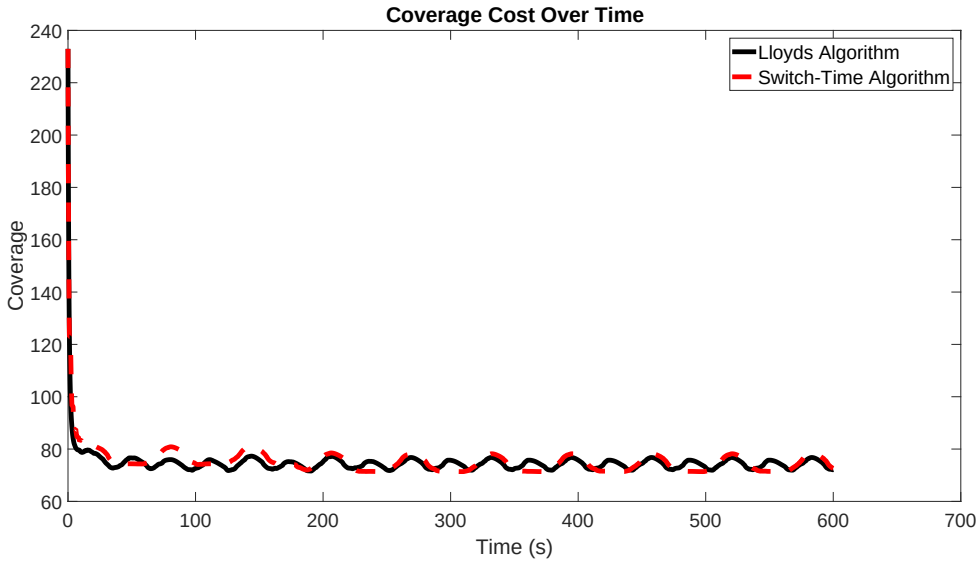

Figure 4.22: The total sum of coverage levels across all agents over time for Lloyd's and the Switch-Time Optimization algorithms using a time varying density distribution.

CHAPTER 5
CONCLUSION

In this thesis, the reduction in energy usage is achieved through compromises in the coverage cost of the multi-robot team. We do not presume to analyze the effects of that compromise on a system, but present a framework to quantify the balance between coverage and energy. The algorithms presented in this thesis will provide the most improvement in scenarios where surveillance must consistently be obtained using agents with finite energy levels. This is achieved through a trade-off between the distance of a robot from the centroid of its Voronoi cell, and the energy required to traverse that distance. We present two separate and distinct strategies that both perform this balance.

We demonstrate that the developed algorithm's allow robots to operate for longer periods of time without recharging, while achieving the desired trade-off between coverage and battery consumption, by comparing the performance against robots executing a canonical coverage control algorithm. In the case of persistent monitoring, the developed algorithms produced a slower rate of energy loss than the canonical Lloyd's algorithm. The relative importance of coverage and energy is left to the designer as a parameter that can be fit to the application under consideration. The developed methods are completely decentralized and only use information already known to the robots while executing the coverage control algorithm.

REFERENCES

[1] G. Antonelli, A. Filippo, and C. Stefano, "The null-space-based behavioral control for autonomous robotic systems," *Intelligent Service Robotics*, vol. 1, pp. 27–39, 1 2008.

[2] S. Sariel, T. Balch, and N. Erdogan, "Naval mine countermeasure missions," *IEEE Robotics Automation Magazine*, vol. 15, pp. 45–52, 2008.

[3] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada, "Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research," *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, vol. 6, pp. 739–743, 1999.

[4] L. C. et al., "Deploying air-ground multi-robot teams in urban environments," *Swarms to Intelligent Automata*, vol. 3, pp. 223–234, 2005.

[5] C. Sotzing and D. Lane, "Improving the coordination efficiency of limitedcommunication multiautonomus underwater vehicle operations using a multiagent architecture," *Journal of Field Robotics*, vol. 27, pp. 412–429, 4 2010.

[6] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, pp. 243–255, 2004.

[7] I. Hussein and D. Stipanovic, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, pp. 642–657, 2007.

[8] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[9] L. Apvrille, T. Tanzi, and J. L. Dugelay, "Autonomous drones for assisting rescue services within the context of natural disasters," *2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS)*, 2014.

[10] C. Haddal and J. Gertler, "Homeland security: Unmanned aerial vehicles and border surveillance," *Congressional Research Service*, 2010.

[11] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques," *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, pp. 492–497, 2005.

[12] J. Derenick, N. Michael, and V. Kumar, "Energy-aware coverage control with docking for robot teams," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3667–3672, 2011.

[13] N. Mathew, S. Smith, and S. Waslander, "Multi-robot rendezvous planning for recharging in persistent tasks," *IEEE Transactions on Robotics*, vol. 31, pp. 128–142, 2015.

[14] P. Antsaklis, "A brief introduction to the theory and applications of hybrid systems," *Special Issue on Hybrid Systems: Theory and Applications*, 2000.

[15] M. Egerstedt, Y. Wardi, and F. Delmotte, "Optimal control of switching times in switched dynamical systems," *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 3, pp. 2138–2143, 2003.

[16] H. Choset, "Coverage for roboticsa survey of recent results," *Annals of Mathematics and Artificial Intelligence*, pp. 113–126, 2001.

[17] H. Jaleel and M. Egerstedt, "Sleep scheduling of wireless sensor networks using hard-core point processes," *American Control Conference (ACC)*, pp. 6642–6647, 2013.

[18] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM Review*, vol. 41, pp. 637–676, 4 2006.

[19] M. Iri, K. Murota, and T. Ohya, "A fast voronoi-diagram algorithm with applications to geographical optimization problems," *Control and Information Sciences*, vol. 59, pp. 273–288, 1984.

[20] H. Axelsson, M. Egerstedt, Y. Wardi, and G. Vachtsevanos, "Algorithm for switching-time optimization in hybrid dynamical systems," *Proceedings of the 2005 IEEE International Symposium*, pp. 256–261, 2005.

[21] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1699–1706, 2017.

[22] S. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multi-robot control using time-varying density functions," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 489–493, 2015.